11/20/00 U.S. PTO JC965

11-24-00

A

JC950 U.S. PTO 09/718879 11/20/00

Please type a plus sign (+) inside this box → ☐

PTO/SB/05 (11-00)
Approved for use through 10/31/2002. OMB 0651-0032
U S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.	41003.P033
First Inventor	Walter G. Bright
Title	A Lossy Method For Compressing Pictures And Videos
Express Mail Label No.	EL605443130US

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231

1. [X] Fee Transmittal Form (e.g., PTO/SB/17) (Submit an original and a duplicate for fee processing)
2. [X] Applicant claims small entity status. See 37 CFR 1.27.
3. [X] Specification [Total Pages 46] (preferred arrangement set forth below)
   - Descriptive title of the invention
   - Cross Reference to Related Applications
   - Statement Regarding Fed sponsored R & D
   - Reference to sequence listing, a table, or a computer program listing appendix
   - Background of the Invention
   - Brief Summary of the Invention
   - Brief Description of the Drawings (if filed)
   - Detailed Description
   - Claim(s)
   - Abstract of the Disclosure
4. [X] Drawing(s) (35 U.S.C. 113) [Total Sheets]
5. Oath or Declaration [Total Pages 3]
   a. [X] Newly executed (original or copy)
   b. [ ] Copy from a prior application (37 CFR 1.63 (d)) (for continuation/divisional with Box 18 completed)
      i. [ ] DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1 33(b).
6. [ ] Application Data Sheet. See 37 CFR 1.76
7. [ ] CD-ROM or CD-R in duplicate, large table or Computer Program (Appendix)
8. Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary)
   a. [ ] Computer Readable Form (CRF)
   b. Specification Sequence Listing on:
      i. [ ] CD-ROM or CD-R (2 copies); or
      ii. [ ] paper
   c. [ ] Statements verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

9. [X] Assignment Papers (cover sheet & document(s))
10. [ ] 37 CFR 3.73(b) Statement (when there is an assignee) [X] Power of Attorney
11. [ ] English Translation Document (if applicable)
12. [ ] Information Disclosure Statement (IDS)/PTO-1449 [ ] Copies of IDS Citations
13. [ ] Preliminary Amendment
14. [X] Return Receipt Postcard (MPEP 503) (Should be specifically itemized)
15. [ ] Certified Copy of Priority Document(s) (if foreign priority is claimed)
16. [ ] Request and Certification under 35 U.S.C. 122 (b)(2)(B)(i). Applicant must attach form PTO/SB/35 or its equivalent.
17. [ ] Other: ....

18. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment, or in an Application Data Sheet under 37 CFR 1.76:

[ ] Continuation [ ] Divisional [ ] Continuation-in-part (CIP) of prior application No.: ________/________

Prior application information: Examiner ________ Group Art Unit: ________

For CONTINUATION OR DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 5b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

19. CORRESPONDENCE ADDRESS

[ ] Customer Number or Bar Code Label 000025943 (Insert Customer No. or Attach bar code label here) or [ ] Correspondence address below

Name	COLUMBIA IP LAW GROUP, LLC				
Address	4900 SW Meadows Road, Suite 109				
City	Lake Oswego	State	Oregon	Zip Code	97035
Country	USA	Telephone	503-534-2800	Fax	503-534-2804

Name (Print/Type)	Aloysius T.C. AuYeung	Registration No. (Attorney/Agent)	35,432
Signature	[signature]	Date	11/20/00

FEE TRANSMITTAL for FY 2001

Patent fees are subject to annual revision.

TOTAL AMOUNT OF PAYMENT ($) 534.00

Complete if Known	
Application Number	Not yet assigned
Filing Date	November 20, 2000
First Named Inventor	Walter G. Bright
Examiner Name	Not ye assigned
Group Art Unit	Not yet assigned
Attorney Docket No.	41003.P033

METHOD OF PAYMENT

1. [X] The Commissioner is hereby authorized to ~~charge indicated fees and~~ credit any overpayments to:

Deposit Account Number: 501569

Deposit Account Name: Columbia IP Law Group, LLC

[X] Charge Any Additional Fee Required Under 37 CFR 1 16 and 1 17

[X] Applicant claims small entity status See 37 CFR 1 27

2. [X] **Payment Enclosed:**

[X] Check [ ] Credit card [ ] Money Order [ ] Other

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
101	710	201	355	Utility filing fee	355.00
106	320	206	160	Design filing fee	
107	490	207	245	Plant filing fee	
108	710	208	355	Reissue filing fee	
114	150	214	75	Provisional filing fee	

SUBTOTAL (1) ($) 355.00

2. EXTRA CLAIM FEES

			Extra Claims		Fee from below		Fee Paid
Total Claims	31	-20** =	11	X	9.00	=	99.00
Independent Claims	4	- 3** =	1	X	40.00	=	40.00
Multiple Dependent						=	

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
103	18	203	9	Claims in excess of 20
102	80	202	40	Independent claims in excess of 3
104	270	204	135	Multiple dependent claim, if not paid
109	80	209	40	** Reissue independent claims over original patent
110	18	210	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 139.00

**or number previously paid, if greater; For Reissues, see above*

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee or oath	
127	50	227	25	Surcharge - late provisional filing fee or cover sheet	
139	130	139	130	Non-English specification	
147	2,520	147	2,520	For filing a request for *ex parte* reexamination	
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	390	216	195	Extension for reply within second month	
117	890	217	445	Extension for reply within third month	
118	1,390	218	695	Extension for reply within fourth month	
128	1,890	228	945	Extension for reply within fifth month	
119	310	219	155	Notice of Appeal	
120	310	220	155	Filing a brief in support of an appeal	
121	270	221	135	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a public use proceeding	
140	110	240	55	Petition to revive - unavoidable	
141	1,240	241	620	Petition to revive - unintentional	
142	1,240	242	620	Utility issue fee (or reissue)	
143	440	243	220	Design issue fee	
144	600	244	300	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	130	123	130	Petitions related to provisional applications	
126	180	126	180	Submission of Information Disclosure Stmt	
581	40	581	40	Recording each patent assignment per property (times number of properties)	40.00
146	710	246	355	Filing a submission after final rejection (37 CFR § 1.129(a))	
149	710	249	355	For each additional invention to be examined (37 CFR § 1 129(b))	
179	710	279	355	Request for Continued Examination (RCE)	
169	900	169	900	Request for expedited examination of a design application	

Other fee (specify) ______________________________

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 40.00

SUBMITTED BY				Complete *(if applicable)*	
Name *(Print/Type)*	Aloysius T.C. AuYeung	Registration No. *(Attorney/Agent)*	35,432	Telephone	503-534-2800
Signature	[signature]			Date	11/20/00

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Attorney's Docket No.: 041003.P033

UNITED STATES PATENT APPLICATION

FOR

A LOSSY METHOD FOR COMPRESSING PICTURES AND VIDEO

Inventor: Walter G. Bright

Prepared By:
Columbia IP Law Group

Mail Correspondence To:
4900 SW Meadows Rd.,
Suite 109
Lake Oswego, OR 97035

Direct Calls To:
(503) 534-2800

"Express Mail" mailing label number: EL605443130US

A LOSSY METHOD FOR COMPRESSING IMAGES AND VIDEO

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention generally relates to techniques for compressing graphic and video data, and in more particular concerns lossy methods for compressing graphics and video.

Background Information

Image and video data compression refers to a process in which the amount of data used to represent images and video is reduced while the quality of the reconstructed image or video satisfies a requirement for a certain application and the complexity of computation involved is affordable for the application. Various compression schemes have been developed over the years, including both lossy and lossless schemes. As their respective names imply, in a lossy compression scheme some aspect of the original data is discarded (lost), because it was deemed unnecessary to the final application, while in a lossless scheme the original data is reproduced when the compressed image or video is decompressed.

Image and video data, like other computer data, is encoded as a series of bits. A typical still color image comprises a plurality of pixels laid out in a grid called a bitmap having multiple bitplanes laid on top of one

another, wherein the number of bitplanes corresponds to the potential color resolution of the image when it is displayed on an output device such as a printer or monitor. For example, an 8-bit(plane) bitmap has a limit of 256 different colors, while a 16-bit(plane) bitmap has a limit of 65536 different colors. In addition to color images, there are black and white images (also known as monochrome, comprising one bit plane), and grayscale bitmaps (typically comprising four or eight bit planes).

In an original bitmap, data corresponding to each pixel of data must occupy a memory space equal to the number of bitplanes in the bitmap. Accordingly, a 1024 x 1024 pixel 24-bit bitmap requires three Mbytes of memory if it was to be stored in its original format. This is clearly excessive, and provided the original motivation for developing compression schemes to reduce the amount of memory required to store images.

When a wider range of colors is needed, it requires mixing a limited number of primary colors. There are two fundamental color mixing schemes, *RGB* and *CMYK*, depending upon the intended output medium. RGB color stand for the red, green, and blue additive primary colors typically used in devices that emit light, such a computer monitors. CMYK color stands for the cyan, magenta, and yellow subtractive primary colors, plus black for contrast, typically used for printed materials. A larger value

for a given primary results in larger spots of that color, applied in a fine pattern on a white page. Each of the three primary inks absorbs certain colors from the white light impinging on the printed page. Because CMYK operates by removing specific colors from white, it is called subtractive color.

An increasingly popular color specification method called *HIS* can be used independently of the mixing scheme. HIS stand for Hue, Saturation, and Intensity. *Hue* is what a person thinks of as the color of an object, such as orange, green or purple. *Saturation* describes the amount of white. The brown color of baking chocolate, for example, is very saturated, while the color of milk chocolate is the same hue, but less saturated. *Intensity* describes the brightness and, for example, is mainly what distinguishes an orange in sunlight form an orange in shade. Also, the word luminosity is often substituted for intensity in which case HSI becomes HSL.

Initially, image data was compressed using schemes originally designed for compressing text data – that is schemes that only considered the binary content and not the qualitative content of the data. These schemes include *run-length compression*, wherein a series of repeated values (i.e., identical adjacent pixels) are replaced by a single value and a count, *Huffman encoding*, which uses shorter codes for frequently

occurring things and longer codes for infrequently occurring things, *LZW* (Lempel, Ziv and Welch) *compressio*n, an adaptive version of Huffman encoding, and *arithmetic compression.* These schemes are all candidates for lossless or near lossless compression. However, they are limited in how much compression they can produce, and are therefore not suitable for many compression requirements. An example of a popular compression scheme that uses a variable-length LZW compression algorithm is the *GIF* (Graphics Interchange Format), which is presently limited to 256 24-bit colors, has no provisions for storing grayscale or color-correction data, and doesn't support the CMYK or HIS models.

The most commonly cited lossy compression scheme is the baseline algorithm of *JPEG* (the Joint Photographic Experts Group of the International Standards Organization (ISO). Although JPEG defines several algorithms, the most commonly referred to as JPEG is the lossy, baseline algorithm. JPEG algorithms begin by separating the chroma (color) information from the luminance (brightness) information, to take advantage of the human eye's tolerance for lower color resolution. Next, the image is broken up into smaller rectangular tiles comprise 8x8 groups of pixels. The lossy algorithm then applies a mathematical technique to the tiles, known as *Discrete Cosine Transform* (DCT). After this transformation, individual pixels no longer exist. Rather, they are

represented by a series of patterns describing how rapidly the pixels vary. Since individual pixel identities are lost in DCT, this form of JPEG is called lossy.

A primary disadvantage with JPEG is that there may be discontinuities at the boundaries of each tile. For example, while JPEG does a fairly good job of providing continuity (i.e., no rapid changes) in adjacent pixels within each tile, there is no provision for blending the pixels of the outlines of the tiles. As a result, some JPEG images, especially those that apply a high compression ratio, look like a set of tiles that are reassembled together to form a composite image. Ideally, a viewer should not perceive that individual tiles are used. This problem is especially pronounced when scaling images. Accordingly, it is desired to provide an image and video compression scheme that provides similar benefits to JPEG compression, but eliminated the discontinuities that are often caused by the compression scheme.

SUMMARY OF THE INVENTION

The present invention provides a lossy compression scheme that addresses many of the limitations found in the prior art by enabling data corresponding to still images and video frames to be compressed and decompressed such that pixel discontinuities are substantially unnoticeable, even when images are scaled. The invention employs a recursive division and comparison scheme, wherein areas of a grid corresponding to the image data are divided into increasingly smaller triangles based on the level of detail contained within each triangle. For instance, areas with a lot of detail will comprise many small triangles, while areas with less detail will have larger triangles. Data is then stored defining each of the triangles and actual and/or predicted component values corresponding to pixels within each triangle. [In one embodiment, the data defining a triangle are data defining the vertices of the triangle.]

According to a first aspect of the invention, a method is provided for compressing data corresponding to still images and video frames comprising a plurality of pixels arranged in a grid. Initially, each pixel will have a one or more component values, such as a grayscale intensity for grayscale images or RGB component values for color images. The grid of pixels is divided into one or more rectangular areas, preferably by using a small number of squares. Each rectangular area is then evaluated

independently of one another in the following recursive manner. The area is divided into a number of triangles splitting the area. For each triangle, a set of predicted component values are determined, based on actual component values at the vertices of the triangle or actual component values on or proximate to the edges of the triangle. Preferably, the predicted component values are determined by interpolating the actual component values at the vertices or along the triangle edges. The predicted component values for the pixels within the triangle are then compared with the corresponding actual component values to see if a similarity threshold is met. In brief, this test is to determine whether the area of the still image or video frame occupied by the triangle when the image is decompressed and rendered is similar to how the same area appears prior to compressing the image, whereby the similarity threshold controls the level of detail that must be maintained. Preferably, the similarity threshold may be specified by a user, or set at a default value. Additionally, the user may specify that one or more particular areas within the image be maintained at a greater level of detail than other areas. Further, the user may also specify that certain "transparent" areas be not described.

If the similarity threshold is met, the processing of a current triangle is complete and the processing of a next triangle begins. If the similarity

threshold is not met, an optional texture mapping determination is performed. In this determination, the component values of the pixels within the triangle are evaluated to see if they from a pattern corresponding to a predefined texture map. If a match is found, the matching texture map is recorded along with the vertices of the triangle. If no match is found, or if the texture mapping determination is not used, the current triangle is divided in half to make two new triangles, preferably along a line connecting the midpoint of the triangle's hypotenuse with its right angle corner. The foregoing process is then repeated for each of the new triangles. As a result, image data are advantageously represented by a number of triangles with no discontinuities from one triangle to its adjacent neighbor. Further, the triangles will be subdivided such that there are many small triangles corresponding to areas in the image that have a lot of detail, while larger triangles may be used for areas that have lesser detail and areas in which texture mapping is applicable.

In accord with further aspects of the invention, a system is provided for implementing the method comprising a processor and memory in which machine instructions are stored that when executed by the processor, perform the functions of the method discussed above. In addition, an article of manufacture is provided having machine instructions stored thereon for implementing the method when the machine

instructions are executable by a suitable system, such as a personal computer.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing aspects and many of the attendant advantages of this invention will become more readily appreciated as the same becomes better understood by reference to the following detailed description, when taken in conjunction with the accompanying drawings, wherein:

FIGURES 1A is a flowchart illustrating the logic used by the present invention when compressing image data corresponding to a grayscale image;

FIGURE 1B is a flowchart illustrating the logic used by the present invention when compressing image data corresponding to a color image;

FIGURE 2 shows how various triangles are created by the present invention when starting with an initial square grid of pixels;

FIGURE 3 depicts an XYZ coordinate axis set for illustrating how image data corresponding to each pixel is maintained, wherein the X and Y axis correspond to the X and Y position of a pixel within an image, and the Z value corresponds to a grayscale or a color component value;

FIGURE 4 shows a grid of pixels wherein the component values for pixels that are on the edge of a triangle are used for interpolating predicted component values of pixels within the triangle;

FIGURE 5A is an exemplary image prior to compression by the present invention;

FIGURE 5B shows an exemplary set of triangles that are produced when compressing the image of FIGURE 5B with the present invention, wherein the triangles are overlaid on the image;

FIGURE 5C shows the exemplary set of triangles apart from the image;

FIGURE 6 is a flowchart illustrating the logic used by the present invention when decompressing an image; and

FIGURE 7 illustrates an exemplary system for implementing the present invention.

DETAILED DESCRIPTION

With reference to the logic flowchart of FIGURES 1A-B, the method for compressing image data in accord with the present invention starts in a block 10 in which the image area (i.e., pixilated grid for a still image or video frame) is divided into one or more rectangular elements, preferably squares. Although rectangles of any shape may be used, using squares will ensure that each triangle will comprise sides coincident with a set of pixels, and that each triangle formed will comprise a right triangle, simplifying the interpolation calculations discussed below. In addition, the squares should be configured such that a relatively small number of non-overlapping squares are used. Further, description of transparent areas may be skipped.

FIGURE 2 shows a 20 x 20 pixel grid 12 comprising four corner points 1-4 and four points within the grid labeled 5-8 corresponding to an exemplary square area created by block 20. For simplicity, an area comprising only a small number of pixels are considered in the following discussion. It will be understood by those skilled in the art that actual still images and video frames may comprise 1000 or more pixels along each axis, and that a given square or rectangular area might comprise sides with similar lengths. Each point in grid 10 is defined by three component values including an X component, a Y component, and a Z component,

wherein the subscripted numbers correspond to that point (e.g., X_1,Y_1,Z_1 corresponds to point 1, etc.). The X and Y components represent the location of each pixel on a planar X-Y grid, such as the grid of a display monitor. As shown in FIGURE 3, the Z component represents a component value of the pixel. As explained in further detail below, for grayscale images a single grayscale intensity component value will be used, while for color images, three component values will generally be used.

In a first recursive iteration, the following process is applied to each square defined above, as indicated by a start loop block 14. In accord with a block 16, each square is divided along a diagonal to form two triangles. For example, grid 12 is divided along a line 18 to form a triangle 20 having vertices at corners 1, 2 and 3, and a triangle 22 having vertices at points 2, 3, and 4. Each of triangles 20 and 22 define a plane in XYZ space, wherein line 18 comprises a shared edge between the planes, and the Z component corresponds to a "height" of the plane.

Next, a determination is made to whether the image is a grayscale or color image in a decision block 24. It is noted that this determination is made only once per image or video frame, and may be made prior to the start of the process, if desired. If the image is grayscale, the logic proceeds to a start loop block 26, in which the following steps are iterated

recursively for each triangle created by block 16 and subsequent triangles defined within the process.

First, a grayscale intensity value is determined at each vertex of the triangle, as provided by a block 28. For example, if triangle 20 is currently being evaluated, grayscale intensity values will be determined for the pixels corresponding to points 1, 2, and 3. A grayscale intensity value for each pixel within the triangle is then interpolated from the vertex values in a block 30. In general, the interpolation will comprises a linear interpolation, but other interpolations may be used as well, depending on the rate of change in the grayscale intensity of the pixels across a given triangle. For example, a second order interpolation may be better suited than a first order interpolation. The order of the interpolation will depend on the actual component values of the pixels. In a block 32, a similarity calculation is then made to determine a relative difference between the actual grayscale intensity values of the pixels within the triangle and the interpolated values by taking summing the squared differences and taking the square root of the sum divided by the area of the triangle. In one embodiment, a second criterion based on the difference between the actual intensity and the interpolated intensity is also employed to determine similarity.

The logic next flows to a decision block 34, wherein a

determination is made to whether a similarity threshold value is exceeded, wherein the threshold value establishes a minimum level of similarity between the grayscale intensity values of the interpolated pixels and the actual values of the pixels. In essence, the purpose of the foregoing similarity calculation and threshold evaluation is to determine if the interpolated intensity values of the pixels within the triangle are close enough to the actual values such that any loss of detail in the area of an image corresponding to the triangle when it is reproduced (i.e., following decompression of the image data) are limited to the selected threshold level. The threshold level is user-controllable or set to a default value, and may be defined as a single value for an entire image, or may be defined at different levels for one or more areas within the image, as explained in further detail below.

For the illustrated embodiment, if the threshold limit is exceeded, the logic next flows to an optional decision block 35 in which a determination is made to whether a texture map may be applied to the triangle. There are many instances in which one or more portions of an image contain a repeated pattern of pixel intensities that create a "textured" appearance in the image. Accordingly, in optional decision block 35 the actual intensity values of the pixels are examined to see if a texture pattern exists by comparing these values with a set of predefined

texture patterns using one or more pattern recognition algorithms that are well-known in the art. If texture mapping is applicable, the predefined texture map pattern is identified along with intensity data in a block 36. The intensity data controls how the texture map is applied during the decompression process described below.

If no match with a predefined pattern is found, the current triangle is divided to create two new triangles in a block 37. Preferably, the triangle is divided by creating a line from the midpoint of the hypotenuse of the triangle to its right angle corner. For example, triangles 20 and 22 were created with dividing line 18, which is the hypotenuse for both triangles 20 and 22 and has a midpoint at a point 5. Accordingly, a dividing line 38 is used to connect point 5 with point 1 (the right angle corner of triangle 20) to form two new triangles 40 and 42. In alternate embodiments, non-equal triangles may be used instead. The logic then flows back to block 26 and the process is repeated again for triangle 40.

Suppose that triangle 40 contains little detail such that the similarity threshold value is not exceeded when triangle 40 is evaluated, or a suitable texture map is identified. Accordingly, the logic will proceed to a decision block 44 in which a determination is made to whether there are any more triangles or squares to process. In this case, triangles 42 and 22 still have to be evaluated, so the answer to decision block 44 is yes,

and the logic proceeds to start block 26, and the processing of triangle 42 begins. Although the order that the triangles are processed is not critical, it is preferable to work on the most-recently divided triangle first.

This time, suppose that triangle 42 contains a moderate level of detail towards its top left-hand corner, resulting in the threshold value being exceeded. As a result, in block 36 triangle 42 is divided by a line 46 extending from a point 6 of the hypotenuse of triangle 42 to point 5, which corresponds with the right angle corner of triangle 42, thereby creating a new pair of triangles 48 and 50 (which for ease of understanding, are equal triangles). The logic then flows back to start block 26 in which triangle 48 is evaluated. Upon evaluation of triangle 48, it is determined that it needs to be divided in half by creating a line 52 from a point 7 (the midpoint of its hypotenuse) to point 6 (its right angle corner), thereby creating triangles 54 and 56. Similarly, upon evaluation of triangle 54, it is also determined that this triangle needs to be divided in half, which is accomplished by creating a line 58 from a point 8 (at the midpoint of the hypotenuse for triangle 54 to point 7 (the right hand corner for triangle 54), creating triangles 60 and 62.

Now suppose that when triangle 62 is evaluated, it is determined that the threshold value is not exceeded. As a result, the answer to decision block 34 is no and the answer to decision block 44 is yes, since

triangles 62, 56, 50 and 22 are still yet to be evaluated. Further suppose that when each of triangles 62, 56, 50 and 22 are evaluated, the similarity threshold value is not exceeded, resulting in no additional triangles being produced. After the evaluation of triangle 22, the answer to decision block 44 is no to triangles (all the triangles in the square (i.e., grid 12) have been evaluated), and no to squares if no more squares are to be processed. If another square needs to be processed, the logic flows back to start block 14, and the preceding process is repeated iteratively until all of the triangles in the new square meet the similarity threshold criteria. At this point, data defining each triangle and actual and/or predicted pixel component values will be stored in memory or on a persistent storage device as a file, preferably in a binary form. In one embodiment, the data defining a triangle are data defining the vertices of the triangle. Further, the data will indicate whether extrapolation is employed, and if employed, the kind of extrapolation employed. For example, suppose that linear interpolation is used for all of the triangles within an image. In this case, information in a header portion of a data file could indicate that linear interpolation was used on all triangles, whereby the only information that would be needed is the XY values of the vertices of each triangle, and the Z component value(s) for the pixels at those vertices. In other instances, the data may include information corresponding to the interpolation of

each triangle, identification of an applicable texture map for the triangle, pixel XY and component values if the edges of triangles are used for interpolation, etc. Thus, under the presentation, image data are advantageously represented by a number of triangles with no discontinuities from one triangle to its adjacent neighbor.

Once processing of all of the squares have been completed and the corresponding data has been stored, the logic flows to a block 64 in which an optional data compression function is performed to reduce the size of a resulting compressed image data. For instance, an initial compressed image file may comprise a list of the vertices of the triangles and corresponding grayscale intensities values that are determined during the processing of all the squares within the current image being compressed. This list typically will be in the form of a string of bits or characters. In addition, if texture mapping is applicable for any of the triangles, the compressed image data will include a string or data structure for each instance of texture mapping identifying the vertices of the triangle and the appropriate pre-defined texture map. These strings may then be compressed in block 64 by one of several well-known data compression algorithms, including Huffman encoding, LZ (Lempel and Ziv) and LZW (Lempel, Ziv and Welch) compression, and arithmetic compression.

Compressing Color Images

The process for compressing color images with the present invention is similar to the process discussed above with respect to compressing grayscale images except that each pixel in a color image comprises three components rather than a single component. For example, color images are typically represented on video displays through use of the red-green-blue (RGB) color model. This is due to the following feature of the human perception of color. The color sensitive area in the human vision system (HVS) consists of three different sets of cones, wherein each set is sensitive to the light of one of the three primary colors: red, green, and blue. Consequently, any color sensed by the HVS can be considered as a particular linear combination of the three primary colors. This result has been shown by numerous studies.

The RGB model is used mainly in color image acquisition and display. For example, in a 24-bit display system, 8 bits is allocated for each of the red, green and blue color components. Additionally, display monitors are designed with three sets of electron guns and corresponding masks, one for each of the primary colors. As a result, it generally is necessary to start with an RGB representation of an image or video frame when compressing these images.

Although images are displayed using the RGB model, in color

signal processing, including image and video compression, various *luminance-chrominance* color models are more efficient and widely used. This has something to do with the color perception of the HVS. It is known that the HVS is more sensitive to green than to red, and is least sensitive to blue. An equal representation of red, green, and blue leads to inefficient data representation when the HVS is the ultimate viewer. Allocating data in accord with the perceptions of the HVS results provides more efficient encoding.

In a luminance-chrominance color models, luminance is concerned with the perceived brightness, while chrominance is related to the perception of hue and saturation of color. Roughly speaking, the luminance-chrominance representation agrees more with the color perception of the HVS. This feature makes luminance-chrominance color models more suitable for color image processing.

Common luminance-chrominance color models that are implemented in imaging systems include the HIS (Hue, Intensity, Saturation) model, the YUV model (used for PAL (Phase Alternating Line) TV systems), the YIQ model (used for NTSC (National Television Systems Committee) TV systems), the YDbDr model (used in the SECAM (Sequential Couleur a Memoire) TV system), and the YCbCr model, which is used for the JPEG and MPEG international coding standards.

Under the YCbCr model, Y corresponds to the luminance, while Cb and Cr correspond to chrominance values. The following equation is used to convert RGB values to YCbCr values.

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} \begin{pmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 16 \\ 128 \\ 128 \end{pmatrix} \qquad (1)$$

It is well-known that the HVS is much more sensitive to the luminance component than to the chrominance components. This leads to a common practice in color image and video encoding: using full resolution for the intensity component, while using a 2 by 1 subsampling both horizontally and vertically for the two chrominance components.

In accord with the foregoing knowledge of the HVS, the present invention preferably implements the YCbCr model when compressing color images, wherein the “Y” (i.e., luminance) component preferably is used in determining whether the similarity threshold is met. It is noted that other luminance-chrominance models may be used, as well as the original RGB data, and that the chrominance components may also be considered in determining whether the similarity threshold is met.

With reference to FIGURE 1B, the process for compressing color images begins in a start loop block 66, which is applied to each triangle produced in a recursive fashion. First, the RGB color component values for the pixels are converted to YCbCr values per equation 1 in a block 68.

While this step is shown within the processing loop of each triangle, this step may be performed on all of the pixels within the image or video frame prior to dividing the image area into squares, or for each square.

The following steps are then applied to each color component, as indicated by a start loop block 70. First, in a block 72 the value of the component is determined for each vertex point. Estimated values for each of the pixels in the triangle are then calculated (preferably by interpolation) using the component values at the vertex points, as provided by a block 74. In a block 76, a similarity comparison calculation is then performed comprising summing the squared differences between the interpolated values and the actual values of the color component for each pixel and taking the square root of the sum and dividing by the area of the triangle. The calculated similarity value is then compared with a threshold value in a decision block 78 to determine whether the triangle needs to be further divided. If the threshold is not exceeded and further color components need to be evaluated for the triangle, the logic loops back from a loop end block 79 to start loop block 70, and the next color component is evaluated using the same process. However, it should be noted that it is preferable to use a lower threshold value for the luminance (Y) component than the chrominance (Yb, Yr) components, since the HVS is more sensitive to luminance. Furthermore, in an alternative

embodiment, only the luminance component needs to be evaluated to determine whether the interpolated pixel values are similar enough to the actual pixel values to meet the threshold requirement.

If the threshold is exceeded for any of the color components (if all three components are evaluated) or for the luminance component (if only the luminance component is evaluated), the logic flows to an optional decision block 80 in which it is determined whether the pixel component values correspond to a predefined texture map, in a manner similar to that described above. If texture mapping is applicable, the texture map data is saved in a block 81, and the logic proceeds to end loop block 79. If texture mapping isn't applicable, the logic flows to a block 82 that divides the triangle from the midpoint of its hypotenuse to its right angle corner, and the logic loops back to start block 66 to begin evaluation of one of the new triangles that is created.

After each of the color components has been evaluated, the logic flows to a decision block 84 in which a determination is made to whether there are any more triangles or squares to process. If the answer is yes, the logic loops back to the appropriate loop start block and processing of the next triangle or square begins. After all of the squares and triangles have been processed, data is generated comprising the X and Y values of each vertex and the color component values for that vertex. If texture

mapping is applicable, the compressed image data will also include separate data strings or data structures for each texture mapping instance. In addition, the data may include information pertaining to the order of interpolation used. As an optional step, the data may be compressed in a block 86 using on of the data compression schemes discussed above.

Pixel Interpolation Based on Edge Data

In some instances, especially when there is a large amount of detail in an image, it may be advantageous to interpolate the pixel component values based on the actual values of the pixels in the edges (i.e., lines) that form the triangles rather than using just the three vertex values. For example, consider a triangle 88 shown in FIGURE 4, which includes a plurality of pixels 90. Each pixel 90 has a grayscale intensity value disposed adjacent to it corresponding to a scale from 0 to 7, wherein 0 represents a minimum intensity level and 7 represents a maximum intensity level.

As depicted in the drawing, there are several bands having different intensity levels that run across triangle 88. For example, each of the pixels with a grayscale intensity value of 6 comprises a portion of a band, while groups of pixels with intensity value of 3 and 1 respectively comprise other bands. Bands of this type are often encountered within

images.

Under the present embodiment, the intensity values for the pixels within the boundary defined by the edges of a given triangle are interpolated based on the values of the pixels in the triangle edges. As a result, banded regions within triangles will generally be stored in the compressed image data such that these regions are accurately reproduced upon decompressing the data. Furthermore, the number of triangles that are required to create an accurate image definition may be sharply reduced, as is shown by the dashed lines in the Figure, wherein these lines represent triangles that would normally have been produced if only the vertex data was used for interpolation.

In addition to interpolation using triangle edge data, it also may be desirable to ensure that individual triangle edges are not recognized when the image data is decompressed. This can be accomplished by having the three vertices of each triangle define a surface by setting the second derivative with respect to the color of a raster line on the image as it crosses the line between to vertices to zero. This will smooth out any abrupt change in the slope, resulting in triangles that are substantially invisible to the human eye, even at the highest, lossiest compression settings.

An application of the present invention when used to compress an

exemplary grayscale image is now described with reference to FIGURES 5A-C. FIGURE 5A shows a still grayscale image 100 of a cat in a doorway, comprising 600 pixels high by 400 pixels wide. FIGURE 5B illustrates an exemplary set of triangles 102 that are created during compression of image 100 by the present invention, wherein the triangles are overlaid on top of the image. FIGURE 5C shows triangle set 102 separately.

In accord with the method described above, image 100 is first divided into a (preferably minimum) number of squares, including a large square 104 comprising 400 x 400 pixels, and two smaller squares 106 and 108, comprising 200 x 200 pixels each. Beginning with square 104, the square is first divided in half along one of its diagonals, and then the similarity comparison threshold evaluation corresponding to blocks 28-36 in FIGURE 1A is performed reiteratively, creating multiple triangles, each subdividing a previous triangular area in half. As can be recognized by examining FIGURE 5B, the density of the triangles is approximately proportional to the level of detail in each area of square 104 (and for square 106 and 108 as well). For example, small triangles are necessary to accurately store data corresponding to the details of the cat and the curves in the gate, while larger triangles may be used for lower detail areas, such as the wall on the right-hand side of the image. Smaller

triangles are also necessary for high-contrast areas, such as the near-vertical lines appearing behind the cat. In addition, where applicable, texture mapping may be performed.

The same process that was used for square 104 is then applied to each of squares 106, resulting in a set of image data in a compressed format. The set of image data may then be compressed using one of the well-known data compression algorithms discussed above.

As discussed above, the method additionally enables a user to define one or more areas within an image that are to be compressed such that the detail in these areas is preserved at a higher level than other areas in the image that are not selected. For example, the user may determine that he wants the cat in FIGURE 5A to be preserved with a higher degree of detail than other areas in the image. Accordingly, the user may draw a box 110 around the cat (through use of an appropriate user interface) to identify to an application program implementing the invention that the user wishes this area of the image to be compressed such that the area's detail is preserved at a higher level than other areas in the drawing. There are several ways this can be implemented, including lowering the threshold value for triangles that are contained within and/or overlap the selected area(s).

Decompressing Image Data

Decompressing the image data to produce an image essentially comprises reversing the steps used in creating the compressed image data discussed above. With reference to FIGURE 6, the decompression process begins in a block 112 in which the image data string(s), data structures and any texture mapping data is decompressed if the same was compressed during the image compression process. Next, a start loop block 114 identifies the start of a process loop that reiteratively generates pixel data for each of the triangles that was compressed during the compression process discussed above, wherein the data for each of the triangles comprises either a set of vertices, data corresponding to the edges of the triangle, or a set of vertices with texture mapping data. If the data corresponds to a color image, then similar processing steps are applied to each color component, as indicated by a start block 116.

In a decision block 118, a determination is made to whether texture mapping is to applied to the triangle. If the answer is yes, than an appropriate predetermined texture mapping pattern is used to create component intensity values for each of the pixels within the triangle, as provided by a block 120. If the answer is no, the interpolated values for each of the pixels in the triangle are determined in a block 122 through use of either the vertices data or triangle edge data and corresponding

interpolation order data.

The logic flows next to a decision block 124 in which a determination is made to whether the data corresponds to a color image. If the answer is yes, the logic reaches an end block 126 that causes the process to loop back to start block 116 and begin processing the next color component if additional color components are left to be evaluated. Once all of the color components have been processed, they are converted from luminance/chrominance values to RGB values in a block 128. Note that if the data was never converted from RGB to luminance/chrominance values this step does not occur.

After the data has been converted into an appropriate component set (e.g. RGB or grayscale), the pixels for the triangle are rendered in a block 130, whereupon the logic reaches an end block 132 that causes the logic to loop back to begin loop 114 and begin processing of the next triangle. After all of the triangles have been processed, rendering of the decompressed image is complete.

The rendering process creates varying effects, depending on the order in which the triangles are rendered. For example, the compressed data may be rearranged so that the largest enclosing triangles are rendered first, followed by the next enclosed interior triangle. As a result, the image will start out appearing fuzzy and will get progressively sharper.

In addition, if various vertices are lost during transmission, the result won't be "holes", "black snow", or "pixilated" images, but rather will produce a loss of detail that will be overlooked by most viewers. Furthermore, the present invention's compression scheme easily scales to create scaled images of any size, without producing the "blocky" images that result when other compression schemes are used to produce scaled images. Scaling may be done by simply multiplying all the x, y values by a scale factor. Images can be easily stretched horizontally or vertically (e.g. like an anthropomorphic TV image) by scaling x, y with different scale factors.

Exemplary System for Implementing the Invention

With reference to FIGURE 7, a generally conventional personal computer 200 is illustrated, which is suitable for use in connection with practicing the present invention. Alternatively, a corresponding workstation on a local area network may be used for executing machine instructions comprising a computer program that causes the present invention to be executed. Personal computer 200 includes a processor chassis 202 in which are mounted a floppy disk drive 204, a hard drive 206, a motherboard populated with appropriate integrated circuits (not shown), and a power supply (also not shown), as are generally well known to those of ordinary skill in the art. A monitor 208 is included for displaying graphics and text generated by software programs that are run

by the personal computer, and for graphically representing images and video frames produced by the present invention. A mouse 210 (or other pointing device) is connected to a serial port (or to a bus port) on the rear of processor chassis 202, and signals from mouse 210 are conveyed to the motherboard to control a cursor on the display and to select text, menu options, and graphic components displayed on monitor 208 by software programs executing on the personal computer, such as a photo editing program that implements the present invention. In addition, a keyboard 212 is coupled to the motherboard for user entry of text and commands that affect the running of software programs executing on the personal computer.

Personal computer 200 also optionally includes a compact disk-read only memory (CD-ROM) drive 214 into which a CD-ROM disk may be inserted so that executable files and data on the disk can be read for transfer into the memory and/or into storage on hard drive 206 of personal computer 200. Other mass memory storage devices such as an optical recorded medium or DVD drive may be included. The machine instructions comprising the software program that causes the CPU to implement the functions of the present invention that have been discussed above will likely be distributed on floppy disks or CD-ROMs (or

other memory media) and stored in the hard drive until loaded into random access memory (RAM) for execution by the CPU.

The above description of illustrated embodiments of the invention is not intended to be exhaustive or to limit the invention to the precise forms disclosed. While specific embodiments of, and examples for, the invention are described herein for illustrative purposes, various equivalent modifications are possible within the scope of the invention, as those skilled in the relevant art will recognize. Accordingly, it is not intended that the scope of the invention in any way be limited by the above description, but instead be determined entirely by reference to the claims that follow.

CLAIMS

What is claimed is:

1. A method for compressing image data corresponding to an image comprising a plurality of pixels defining a grid, each pixel having at least one component value, comprising:

dividing the grid into at least one rectangular area;

for each rectangular area:

dividing the rectangular area into a number of triangles, each triangle defining a boundary comprising three edges;

for each of triangle:

identifying the vertices of the triangle;

determining predicted pixel component values for at least a portion of the pixels enclosed within and/or on the boundary of the triangle;

comparing the predicted pixel component values with actual values of said at least one component value to determine if a similarity threshold is met;

processing a next triangle if the similarity threshold is met, otherwise,

dividing the triangle into two new triangles, each defining a boundary and comprising three edges; and

reiteratively repeating identifying the vertices, predicting pixel component values, and comparing actual and predicted pixel component values to determine if a similarity threshold is met for each existing triangle and any new triangles that are created; and

generating compressed image data defining each triangle that is created and actual and predicted pixel component values within the triangle.

2. The method of claim 1, wherein the compressed image data comprises at least one string, the method further comprising compressing said at least one string using a data compression algorithm.

3. The method of claim 1, wherein at least a portion of the triangles are defined by data identifying pixels coincident with or proximate to a set of vertices for the triangle, and the predicted pixel component values for those triangles are determined by interpolating actual pixel component values at the vertices of each triangle.

4. The method of claim 1, wherein the predicted pixel component values are determined by interpolating actual component values corresponding to pixels that lie on and/or proximate to the edges of each triangle.

5. The method of claim 1, wherein the image is a grayscale image, and said at least one component value comprises a grayscale intensity level.

6. The method of claim 1, wherein the image comprises a color image, and said at least one component value comprises three color component values.

7. The method of claim 6, wherein the three color component values comprise a Red component value, a Green component value, and a Blue component value, further comprising converting the Red, Green, and Blue component values into luminance/chrominance component values.

8. The method of claim 7, wherein said comparing of the predicted pixel component values to determine if the similarity threshold is met comprises giving a similarity determination that compares predicted and actual luminance component values greater weight than a similarity determination that considers predicted and actual chrominance component values.

9. The method of claim 1, further comprising:

determining if a texture map can be applied to pixels of a given triangle to meet the similarity threshold; and

storing data identifying the pixels within and/or on the triangle boundary and data corresponding to the texture map for any triangle for which it is determined that texture mapping can be applied.

10. The method of claim 1, wherein each of said at least one rectangle comprises a square.

11. The method of claim 10, wherein the image comprises a plurality of pixels contained within a rectangular grid, and wherein the rectangular grid is divided into a number of non-overlapping squares that contain all of the pixels within the rectangular grid.

12. The method of claim 1, wherein at least one triangle comprises a right angle corner and a hypotenuse, and where dividing said at least one triangle into two triangles comprises dividing said at least one triangle along a line connecting a midpoint of the hypotenuse to the right angle corner.

13. A method for reproducing an image based on a set of compressed image data corresponding to an original image comprising a plurality of pixels defining a grid, said compressed image data including

data defining a plurality of triangles and pixel component values corresponding to each of said plurality of triangles, comprising:

extracting a set of vertices for each of said plurality of triangles;

determining component values of pixels within each triangle;

rendering each triangle in accord with the set of vertices and component values determined for that triangle.

14. The method of claim 13, wherein the compressed image data includes pixel component values for pixels coincident with or proximate to each vertex, and the pixel component values corresponding to other pixels within each triangle are determined as a function of the pixel component values at the vertices of the triangle.

15. The method of claim 14, wherein the pixel component values corresponding to the other pixels within each triangle are determined by interpolating the pixel component values at the vertices of the triangle.

16. The method of claim 13, wherein the compressed image data includes data pertaining to sets of pixels defining edges of at least a portion of said plurality of triangles and including pixel component values for those pixels, further wherein the pixel component values for the triangles are determined as a function of the pixel component values corresponding to the pixels defining the edges of the triangles.

17. The method of claim 13, wherein the compressed image data includes texture mapping data, further including:

determining any triangles from among said plurality of triangles to which texture mapping is to be applied; and

applying texture mapping to the pixels contained within those triangles.

18. The method of claim 13, wherein the compressed image data corresponds to a color image and includes pixel component data in accordance with a luminance/chrominance color model, further comprising converting the luminance/chrominance color model data to red, green and blue (RGB) color components for each pixel.

19. The method of claim 13, further comprising rendering said plurality of triangles such that enclosing larger triangles are rendered prior to enclosed smaller triangles.

20. A system for compressing image data corresponding to an image comprising a plurality of pixels defining a grid, each pixel having at least one component value, comprising:

a memory in which machine instructions are stored; and

a processor coupled to the memory for executing the machine instructions, said processor implementing a plurality of functions when executing the machine instructions, including:

dividing the grid into at least one rectangular area;

for each rectangular area:

dividing the rectangular area into a number of triangles, each triangle defining a boundary comprising three edges;

for each of triangle:

identifying the vertices of the triangle;

determining predicted pixel component values for at least a portion of the pixels enclosed within and/or on the boundary of the triangle;

comparing the predicted pixel component values with actual values of said at least one component value to determine if a similarity threshold is met;

processing a next triangle if the similarity threshold is met, otherwise,

dividing the triangle into two new triangles, each defining a boundary and comprising three edges; and

reiteratively repeating identifying the vertices, predicting pixel component values, and comparing actual and predicted pixel component values for each existing triangle and any new triangles that are created; and

generating compressed image data defining each triangle that is created and actual and predicted pixel component values within the triangle.

21. The system of claim 20, wherein at least a portion of the triangles are defined by data identifying pixels coincident with or proximate to a set of vertices for the triangle, and the predicted pixel component values are determined by interpolating actual pixel component values at the vertices of each triangle.

22. The system of claim 20, wherein the predicted pixel component values are determined by interpolating actual component values corresponding to pixels that lie on and/or proximate to the edges of each triangle.

23. The system of claim 20, wherein the image comprises a color image, and said at least one component value comprises a Red component value, a Green component value, and a Blue component value, and wherein execution of the machine instructions by the processor further implements the function of converting the Red, Green, and Blue component values into luminance/chrominance component values.

24. The system of claim 20, wherein execution of the machine instructions by the processor further implements the functions of:

determining if a texture map can be applied to pixels of a given triangle to meet the similarity threshold; and

storing data identifying the pixels within and/or on the triangle boundary and data corresponding to the texture map for any triangle for which it is determined that texture mapping can be applied.

25. The system of claim 20, wherein the image comprises a plurality of pixels contained within a rectangular grid, and wherein the rectangular grid is divided into a minimum number of non-overlapping squares that contain all of the pixels within the rectangular grid.

26. An article of manufacture for compressing image data corresponding to an image comprising a plurality of pixels defining a grid, each pixel having at least one component value, comprising:

a memory media adapted to be used with a computer; and

a plurality of machine instructions stored on the memory media, said machine instructions effecting a plurality of functions when executed by the computer, including:

dividing the grid into at least one rectangular area;

for each rectangular area:

dividing the rectangular area into a number of triangles, each triangle defining a boundary comprising three edges;

for each of triangle:

identifying the vertices of the triangle;

determining predicted pixel component values for at least a portion of the pixels enclosed within and/or on the boundary of the triangle;

comparing the predicted pixel component values with actual values of said at least one component value to determine if a similarity threshold is met;

processing a next triangle if the similarity threshold is met, otherwise,

dividing the triangle into two new triangles, each defining a boundary and comprising three edges; and

reiteratively repeating identifying the vertices, predicting pixel component values, and comparing actual and predicted pixel component values for each existing triangle and any new triangles that are created; and

generating compressed image data defining each triangle that is created and actual and predicted pixel component values within the triangle.

27. The article of manufacture of claim 26, wherein at least a portion of the triangles are defined by data identifying pixels coincident with or proximate to a set of vertices for the triangle, and the predicted pixel component values are determined by interpolating actual pixel component values at the vertices of each triangle.

28. The article of manufacture of claim 26, wherein the predicted pixel component values are determined by interpolating actual component values corresponding to pixels that lie on and/or proximate to the edges of each triangle.

29. The article of manufacture of claim 26, wherein the image comprises a color image, and said at least one component value comprises a Red component value, a Green component value, and a Blue component value, and wherein said functions effectuated when executed by the computer further include the function of converting the Red, Green, and Blue component values into luminance/chrominance component values.

30. The article of manufacture of claim 26, wherein said functions effectuated when executed by the computer further include the functions of:

determining if a texture map can be applied to pixels of a given triangle to meet the similarity threshold; and

storing data identifying the pixels within and/or on the triangle boundary and data corresponding to the texture map for any triangle for which it is determined that texture mapping can be applied.

31. The article of manufacture of claim 26, wherein the image comprises a plurality of pixels contained within a rectangular grid, and wherein the rectangular grid is divided into a minimum number of non-overlapping squares that contain all of the pixels within the rectangular grid.

ABSTRACT OF THE DISCLOSURE

A recursive scheme is employed to compress image data, wherein areas of a pixel grid corresponding to the image data are divided into increasingly smaller triangles based on the level of detail contained within each triangle. Data is stored defining each of the triangles and actual/predicted component values corresponding to pixels within each triangle. For each triangle, a set of predicted component values are determined, based on actual component values at the vertices of the triangle or actual component values on or proximate to the edges of the triangle. The predicted component values for the pixels are compared with the corresponding actual component values to see if a similarity threshold is met. If met, the processing of a current triangle is complete and the processing of a next triangle begins. If not, the current triangle is divided, and the foregoing process is then repeated. Texture mapping may also be applied.

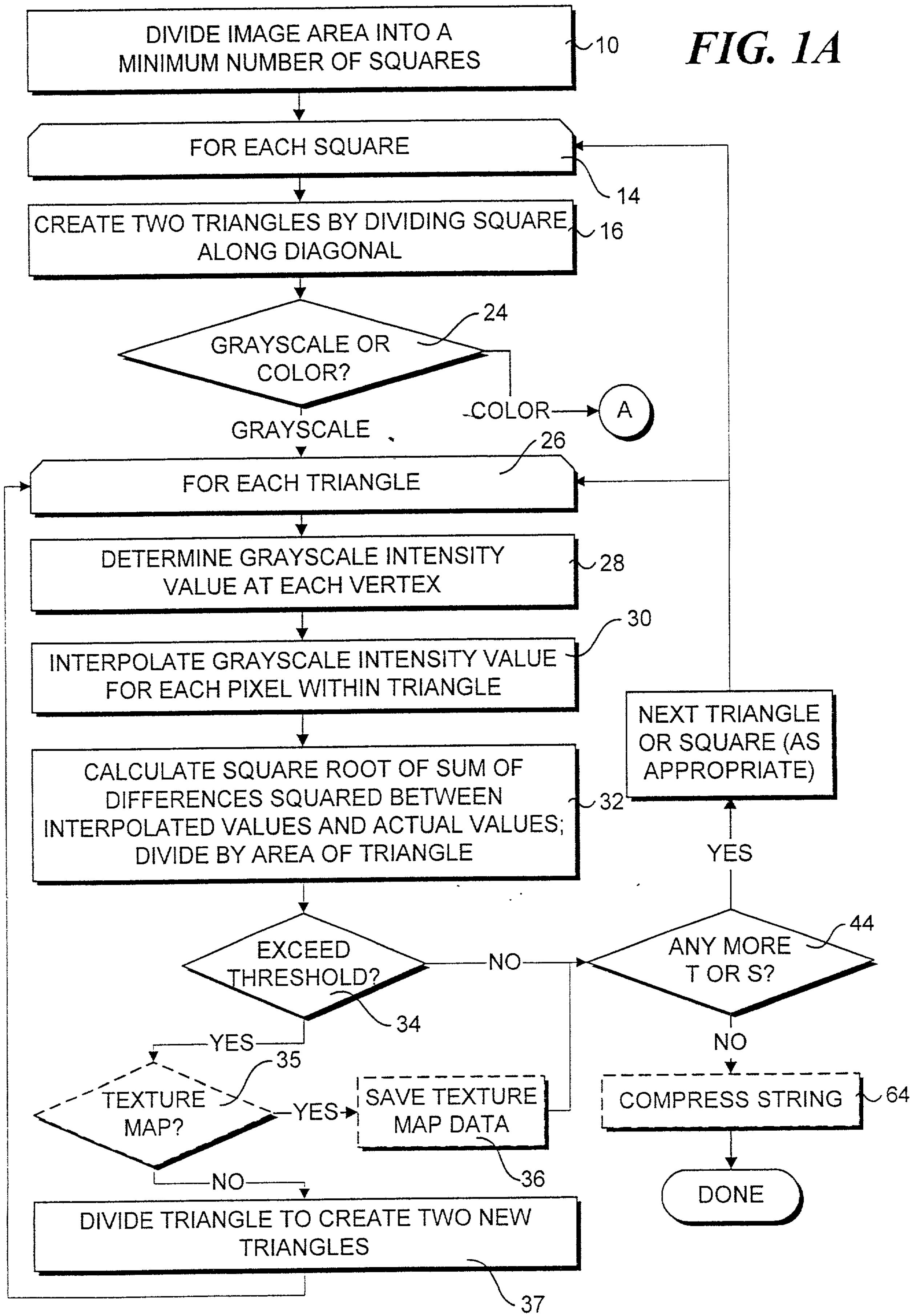

DIVIDE IMAGE AREA INTO A MINIMUM NUMBER OF SQUARES
10
FOR EACH SQUARE
14
CREATE TWO TRIANGLES BY DIVIDING SQUARE ALONG DIAGONAL
16
GRAYSCALE OR COLOR?
24
COLOR
A
GRAYSCALE
FOR EACH TRIANGLE
26
DETERMINE GRAYSCALE INTENSITY VALUE AT EACH VERTEX
28
INTERPOLATE GRAYSCALE INTENSITY VALUE FOR EACH PIXEL WITHIN TRIANGLE
30
CALCULATE SQUARE ROOT OF SUM OF DIFFERENCES SQUARED BETWEEN INTERPOLATED VALUES AND ACTUAL VALUES; DIVIDE BY AREA OF TRIANGLE
32
NEXT TRIANGLE OR SQUARE (AS APPROPRIATE)
YES
EXCEED THRESHOLD?
34
NO
ANY MORE T OR S?
44
YES
TEXTURE MAP?
35
YES
SAVE TEXTURE MAP DATA
36
NO
COMPRESS STRING
64
DONE
NO
DIVIDE TRIANGLE TO CREATE TWO NEW TRIANGLES
37

FIG. 1A

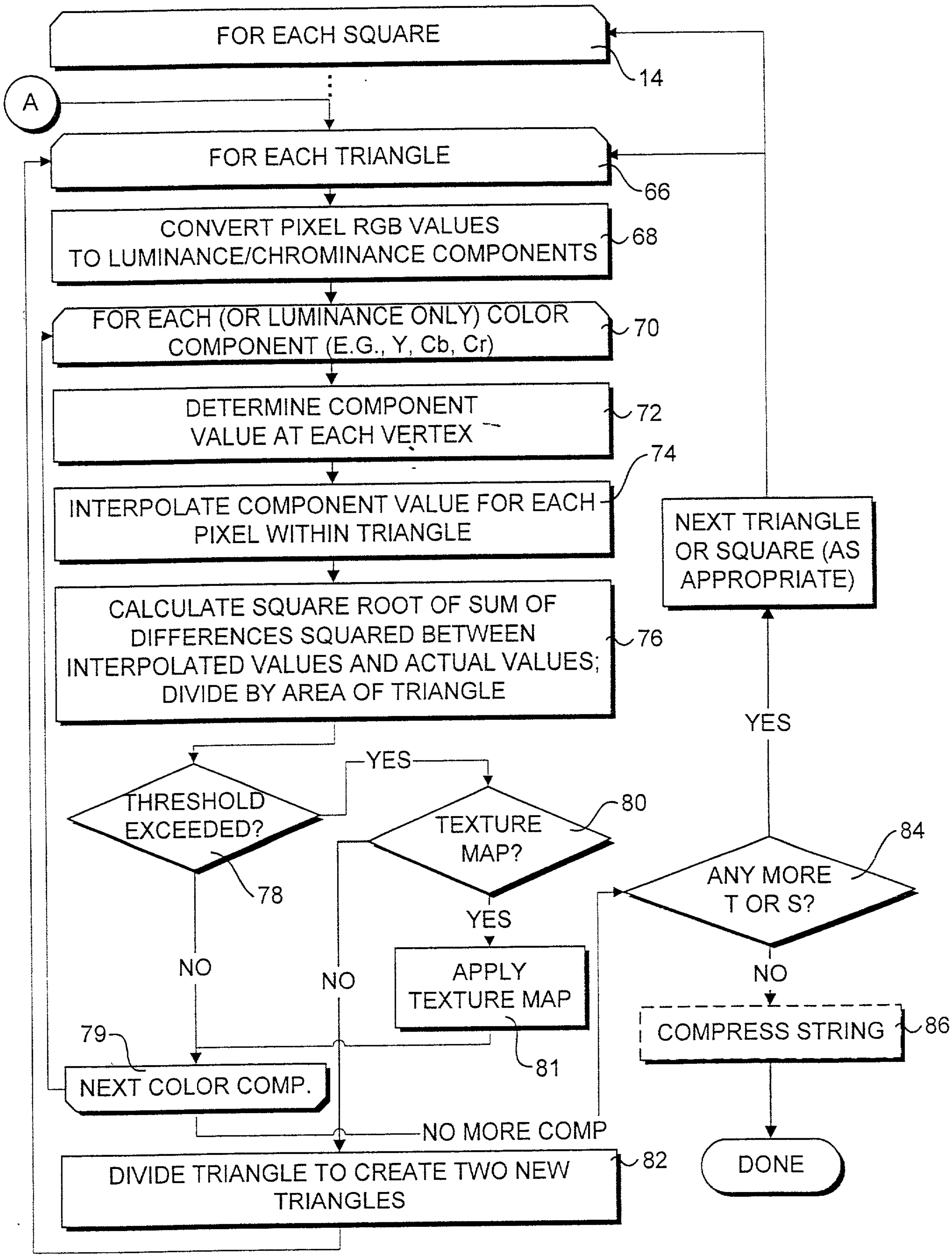
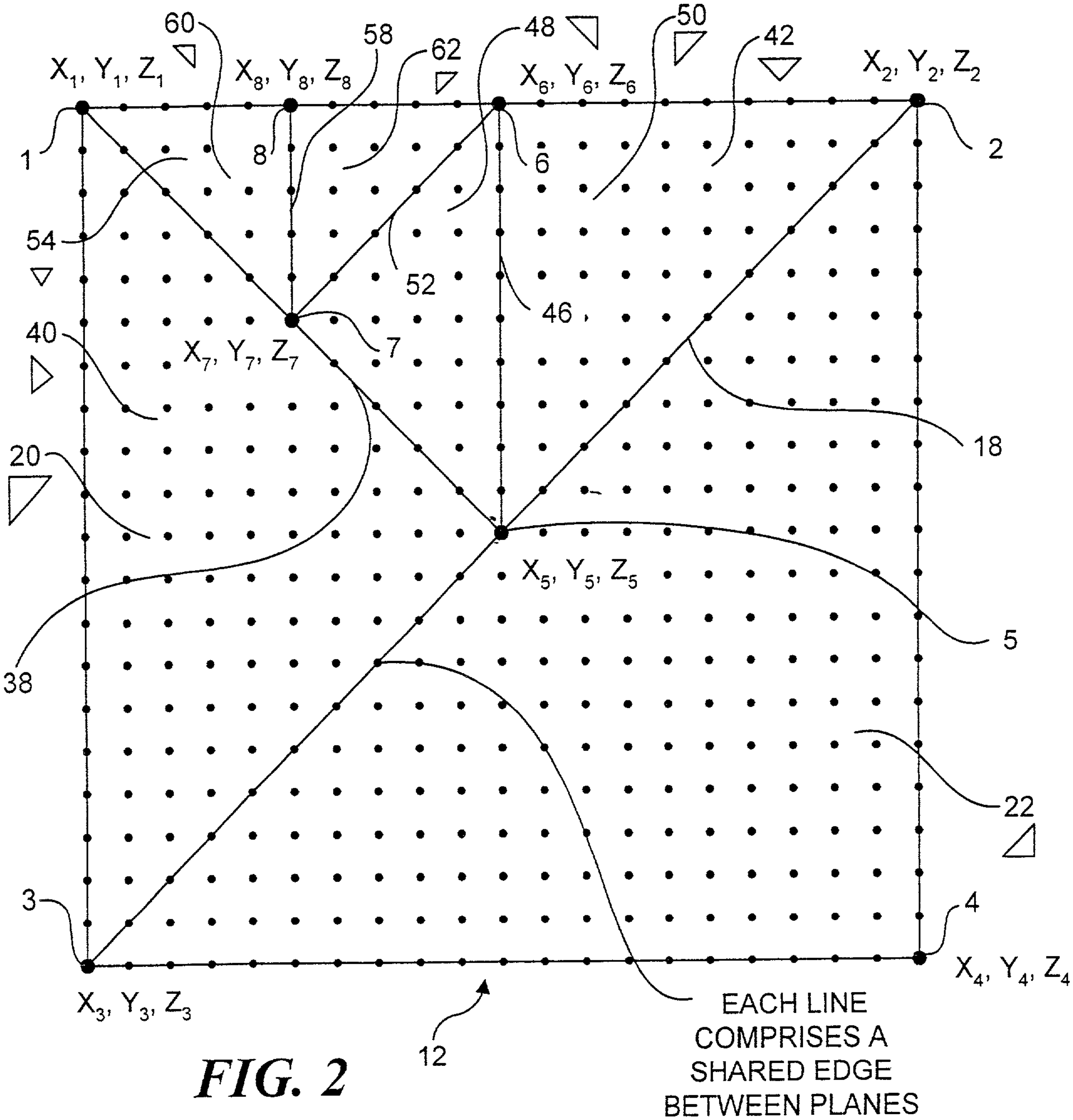
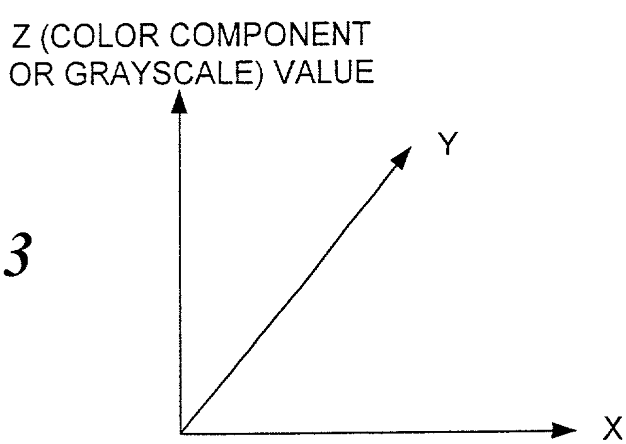

FOR EACH SQUARE
14
A
FOR EACH TRIANGLE
66
CONVERT PIXEL RGB VALUES TO LUMINANCE/CHROMINANCE COMPONENTS
68
FOR EACH (OR LUMINANCE ONLY) COLOR COMPONENT (E.G., Y, Cb, Cr)
70
DETERMINE COMPONENT VALUE AT EACH VERTEX
72
INTERPOLATE COMPONENT VALUE FOR EACH PIXEL WITHIN TRIANGLE
74
CALCULATE SQUARE ROOT OF SUM OF DIFFERENCES SQUARED BETWEEN INTERPOLATED VALUES AND ACTUAL VALUES; DIVIDE BY AREA OF TRIANGLE
76
NEXT TRIANGLE OR SQUARE (AS APPROPRIATE)
YES
THRESHOLD EXCEEDED?
78
YES
TEXTURE MAP?
80
ANY MORE T OR S?
84
NO
YES
NO
APPLY TEXTURE MAP
81
NO
COMPRESS STRING
86
79
NEXT COLOR COMP.
NO MORE COMP
DIVIDE TRIANGLE TO CREATE TWO NEW TRIANGLES
82
DONE

FIG. 1B



X1, Y1, Z1
X8, Y8, Z8
X6, Y6, Z6
X2, Y2, Z2
X7, Y7, Z7
X5, Y5, Z5
X3, Y3, Z3
X4, Y4, Z4
60
58
62
48
50
42
1
8
6
2
54
52
46
7
40
18
20
5
38
22
3
4
12
EACH LINE
COMPRISES A
SHARED EDGE
BETWEEN PLANES

FIG. 2



Z (COLOR COMPONENT
OR GRAYSCALE) VALUE
Y
X

FIG. 3

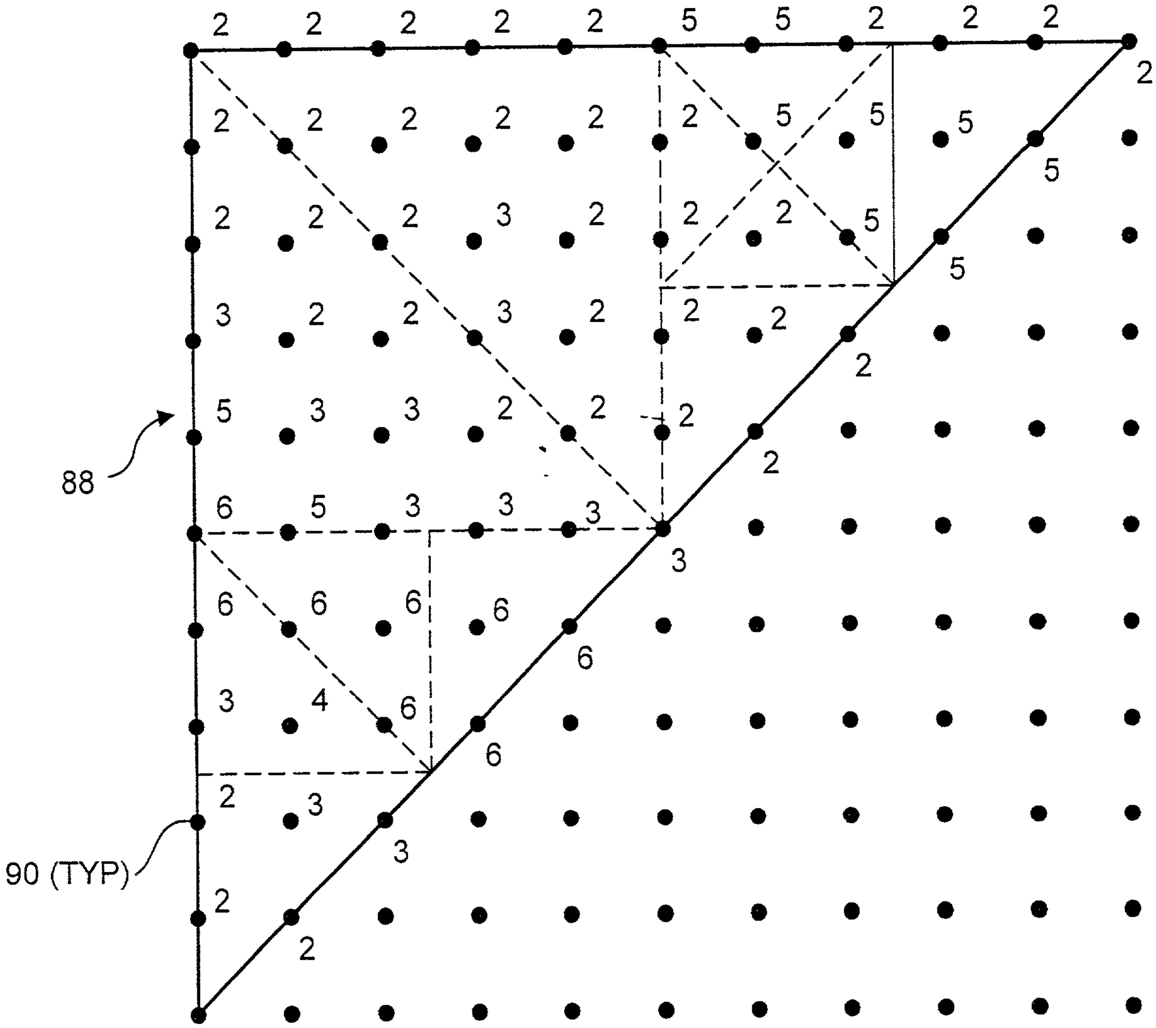


FIG. 4

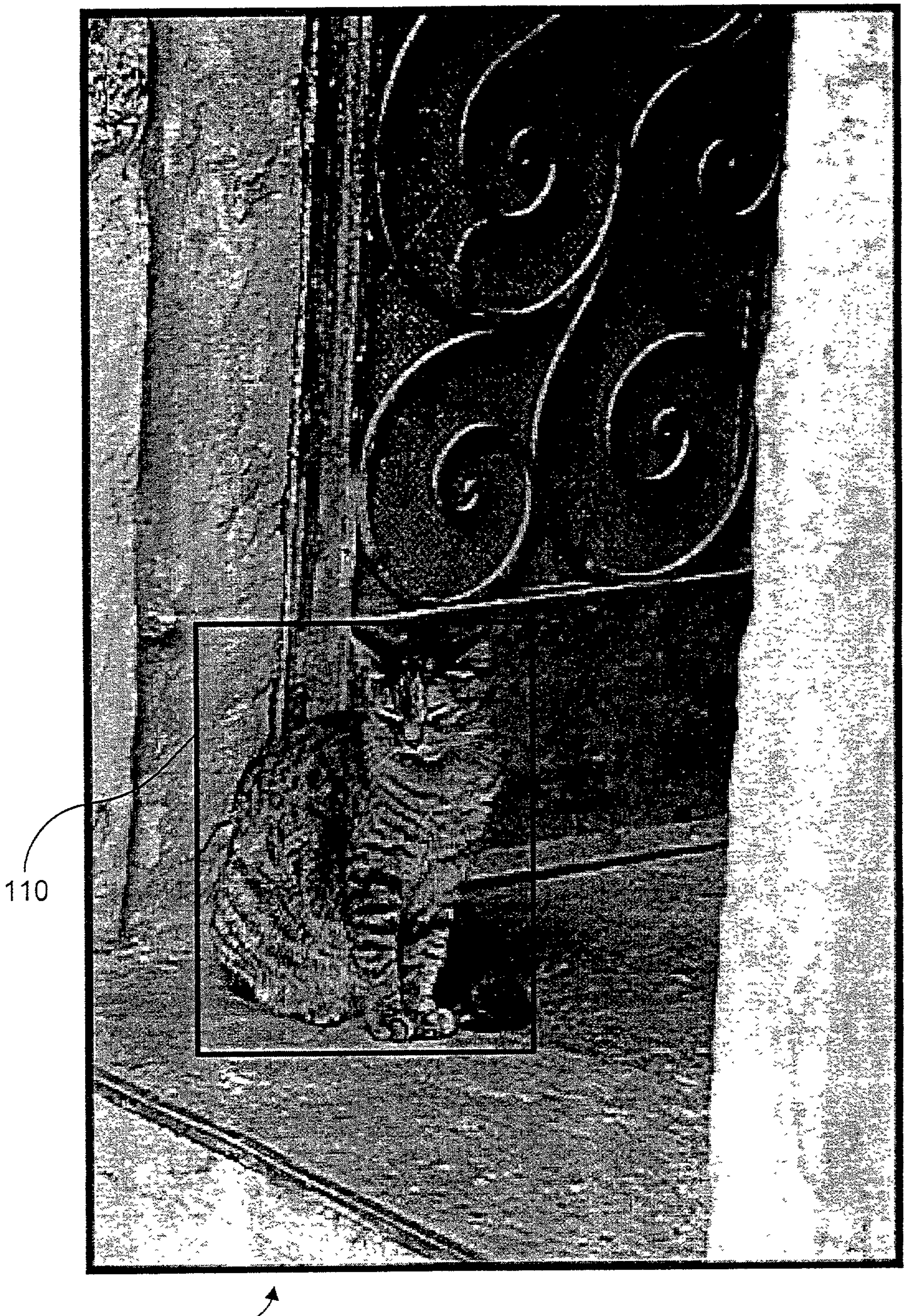


FIG. 5A

100, 102

FIG. 5B

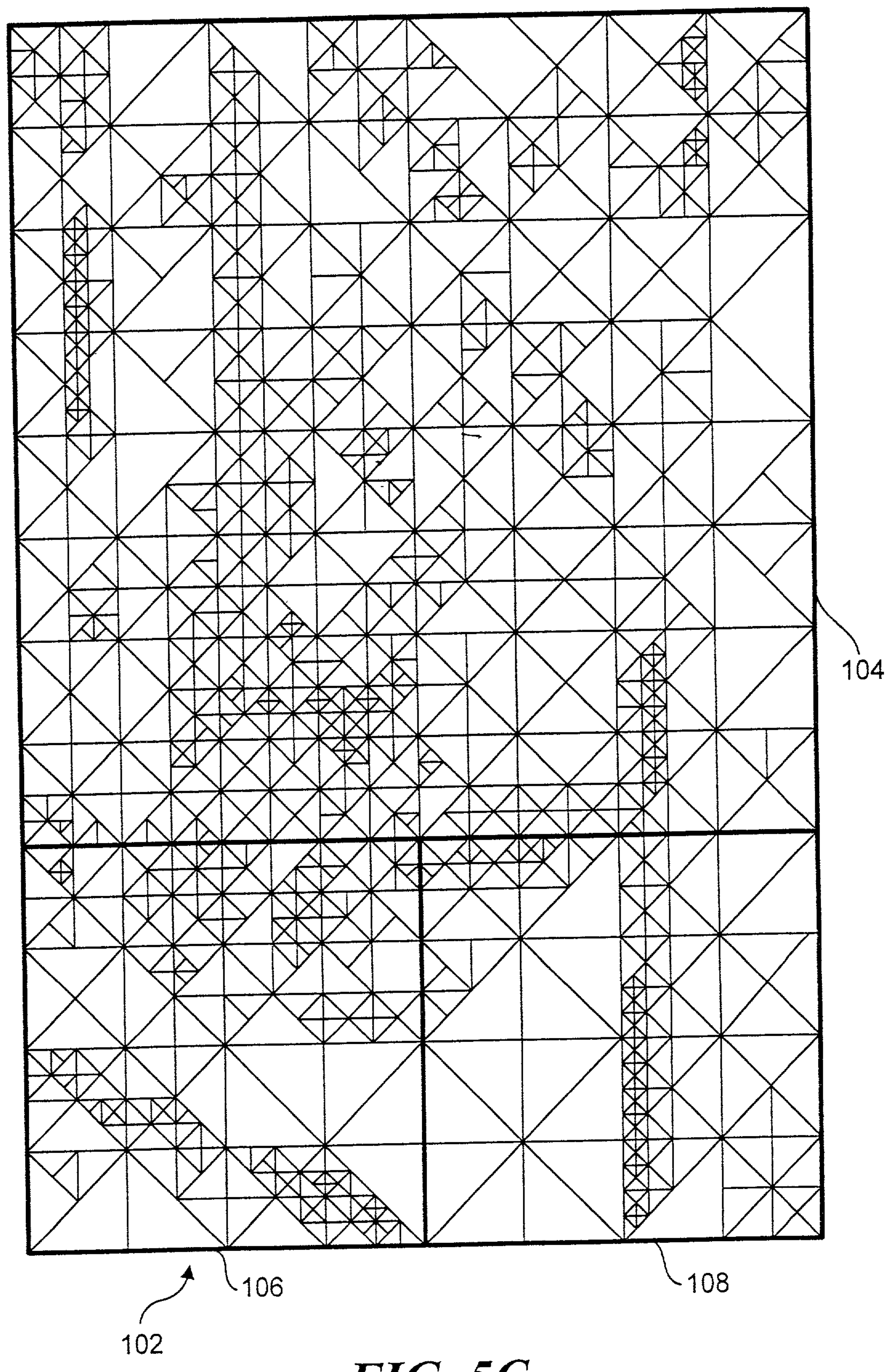
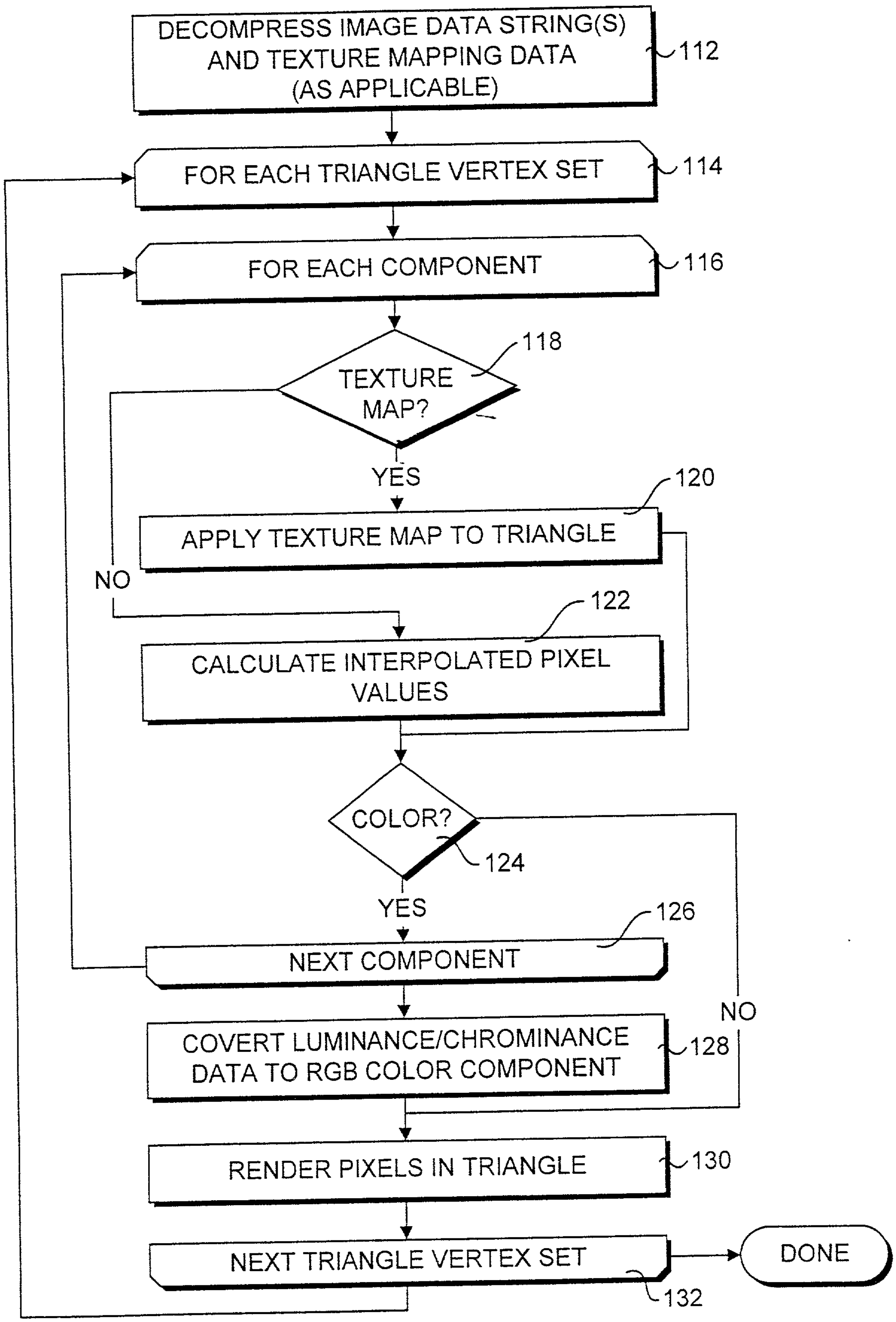


FIG. 5C



DECOMPRESS IMAGE DATA STRING(S) AND TEXTURE MAPPING DATA (AS APPLICABLE)
112
FOR EACH TRIANGLE VERTEX SET
114
FOR EACH COMPONENT
116
TEXTURE MAP?
118
NO
YES
APPLY TEXTURE MAP TO TRIANGLE
120
CALCULATE INTERPOLATED PIXEL VALUES
122
COLOR?
124
YES
NO
NEXT COMPONENT
126
COVERT LUMINANCE/CHROMINANCE DATA TO RGB COLOR COMPONENT
128
RENDER PIXELS IN TRIANGLE
130
NEXT TRIANGLE VERTEX SET
132
DONE

FIG. 6

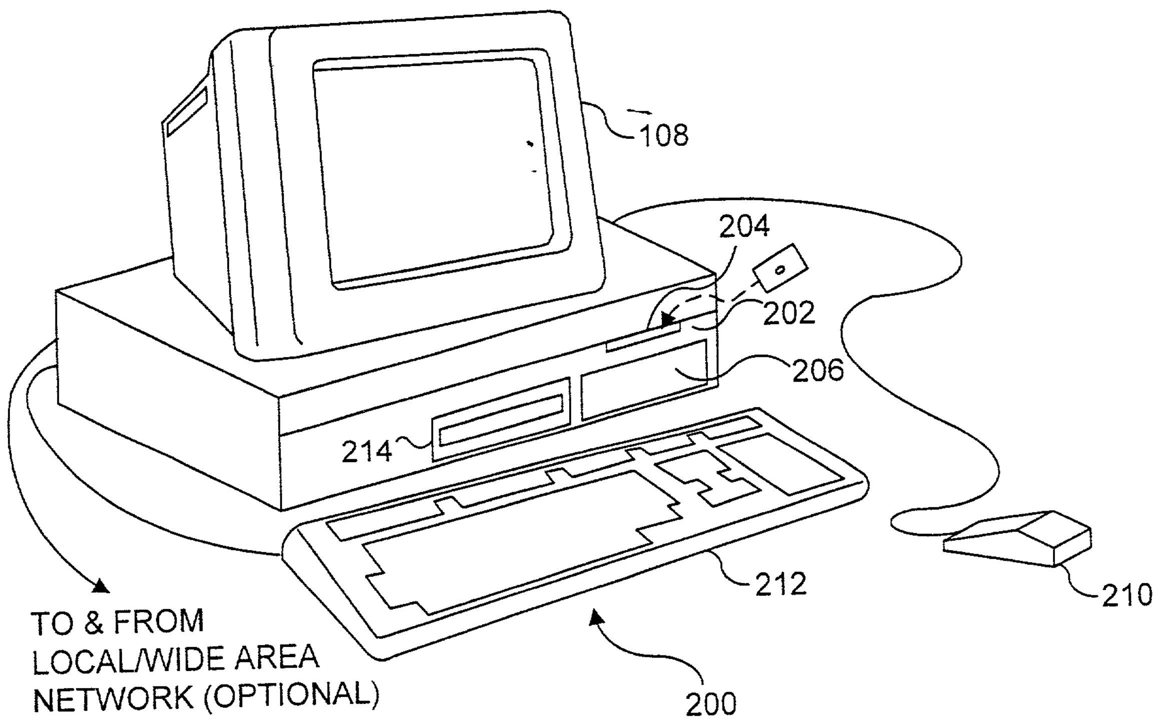

108
204
202
206
214
212
210
200
TO & FROM
LOCAL/WIDE AREA
NETWORK (OPTIONAL)

FIG. 7

Attorney's Docket No.: 0041003.P033 PATENT

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below, next to my name.

I believe I am the original, first, and sole inventor (if only one name is listed below) or an original, first, and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

A LOSSY METHOD FOR COMPRESSING PICTURES AND VIDEOS

the specification of which

X is attached hereto.
___ was filed on ______________________ as
United States Application Number ______________
or PCT International Application Number ______________
and was amended on ______________________.
(if applicable)

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claim(s), as amended by any amendment referred to above.

I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d), of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)			Priority Claimed	
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No

I hereby claim the benefit under title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below

(Application Number)	Filing Date
(Application Number)	Filing Date

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Number)	Filing Date	(Status -- patented, pending, abandoned)
(Application Number)	Filing Date	(Status -- patented, pending, abandoned)

I hereby appoint Aloysius T. C. AuYeung, Reg. No. 35,432; Robert Andrew Diehl, Reg. No. 40,992; Jason K. Klindtworth, Reg. No. P47,211, Robert T. Watt, Reg. No 45,890; as my patent attorney/agent; with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith.

Send correspondence to Aloysius T.C. AuYeung (Name of Attorney or Agent), Columbia IP Law Group, LLC, 4900 S.W. Meadows Rd., Suite 109, Lake Oswego, OR 97035 and direct telephone calls to Aloysius T.C. AuYeung, 503-534-2800 (Name of Attorney or Agent)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Sole/First Inventor Walter G. Bright

Inventor's Signature Walter Bright Date Nov. 20, 2000

Residence Kirkland, WA (City, State) Citizenship U.S.A. (Country)

Post Office Address 550 Kirkland Way,
Kirkland, WA 98033

Title 37, Code of Federal Regulations, Section 1.56
Duty to Disclose Information Material to Patentability

(a) A patent by its very nature is affected with a public interest. The public interest is best served, and the most effective patent examination occurs when, at the time an application is being examined, the Office is aware of and evaluates the teachings of all information material to patentability. Each individual associated with the filing and prosecution of a patent application has a duty of candor and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section. The duty to disclosure information exists with respect to each pending claim until the claim is cancelled or withdrawn from consideration, or the application becomes abandoned. Information material to the patentability of a claim that is cancelled or withdrawn from consideration need not be submitted if the information is not material to the patentability of any claim remaining under consideration in the application There is no duty to submit information which is not material to the patentability of any existing claim. The duty to disclosure all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in a patent was cited by the Office or submitted to the Office in the manner prescribed by §§1.97(b)-(d) and 1.98. However, no patent will be granted on an application in connection with which fraud on the Office was practiced or attempted or the duty of disclosure was violated through bad faith or intentional misconduct The Office encourages applicants to carefully examine:

(1) Prior art cited in search reports of a foreign patent office in a counterpart application, and

(2) The closest information over which individuals associated with the filing or prosecution of a patent application believe any pending claim patentably defines, to make sure that any material information contained therein is disclosed to the Office.

(b) Under this section, information is material to patentability when it is not cumulative to information already of record or being made or record in the application, and

(1) It establishes, by itself or in combination with other information, a prima facie case of unpatentability of a claim; or

(2) It refutes, or is inconsistent with, a position the applicant takes in·

(i) Opposing an argument of unpatentability relied on by the Office, or

(ii) Asserting an argument of patentability.

A prima facie case of unpatentability is established when the information compels a conclusion that a claim is unpatentable under the preponderance of evidence, burden-of-proof standard, giving each term in the claim its broadest reasonable construction consistent with the specification, and before any consideration is given to evidence which may be submitted in an attempt to establish a contrary conclusion of patentability.

(c) Individuals associated with the filing or prosecution of a patent application within the meaning of this section are:

(1) Each inventor named in the application;

(2) Each attorney or agent who prepares or prosecutes the application; and

(3) Every other person who is substantively involved in the preparation or prosecution of the application and who is associated with the inventor, with the assignee or with anyone to whom there is an obligation to assign the application.

(d) Individuals other than the attorney, agent or inventor may comply with this section by disclosing information to the attorney, agent, or inventor.

Each Inventor: Please Sign and Date Below:

Nov. 20, 2000 Date — Walter G Bright — Name: Walter G. Bright

________, 2000 Date — Name:

________, 2000 Date — Name:

________, 2000 Date — Name:

Each Inventor: Please also list the date that you signed the accompanying DECLARATION AND POWER OF ATTORNEY:

Nov. 20, 2000 Date

________, 2000 Date

________, 2000 Date

________, 2000 Date

09725975 .112000